

1. A computer programming method comprising:  
describing data types as abstract without default or implicit implementations and  
distinctly from classes or interfaces;  
describing subtype and supertype relationships between the types; and  
5 describing representations of values of the types as states of classes of objects with  
interfaces.

2. The computer programming method of claim 1 further comprising:  
describing software-visible physical objects and an instruction set using object-oriented  
classes; and  
10 identifying when classes are implemented by a computer.

3. The computer programming method of claim 1 further comprising:  
describing multiple classes of pointer objects, said pointer objects capable of signifying  
objects.

4. The computer programming method of claim 1 further comprising:  
15 describing commands for transferring program control in non-sequential ways as  
routines.

5. The computer programming method of claim 1 further comprising:  
describing at least one of interfaces, classes, enumerations, subroutines, and other  
routines as classes of objects.

20 6. The computer programming method of claim 1 further comprising:  
invoking statements in a compilation with results thereof being incorporated into an  
output of a compiler.

7. The computer programming method of claim 1 further comprising:  
invoking statements in a compilation to interpret literals in an input of a compiler.

8. The computer programming method of claim 1 further comprising:

deriving variable classes from a constant class; and

5 describing one of the variable classes and the constant class using a single descriptor.

9. The computer programming method of claim 1 further comprising:

deriving variable interfaces from a constant interface; and

describing one of the variable interfaces and the constant interface using a single  
descriptor.

10 10. The computer programming method of claim 1 further comprising:

describing formal arguments to routines as objects.

11. The computer programming method of claim 1 further comprising:

describing formal arguments to routines as a number of arguments including type,

interface, or class of each argument, dataflow attribute of each argument, and preconditions and  
15 postconditions of routines.

12. The computer programming method of claim 1 further comprising:

describing subroutines as parameterized classes.

13. A method of compilation comprising:

a. generating a description of a computer architecture as a first library, the

20 description including software-visible objects and the instruction set;

b. implementing a second library of high level objects using the first library; and

c. binding a source program to implementations in the second library to produce  
machine instructions dependent on the computer architecture.

14. A method of re-targeting comprising:

- a. generating a first description of a first computer architecture as a first library, the first description including software-visible objects and instruction set therefor;
- b. generating a second description of a second computer architecture as a second library, the second description including software-visible objects and instruction set therefor;
- c. implementing the first description using the second library to produce a third library; and
- d. binding a source program to implementations in the third library to produce machine instructions dependent on the second computer architecture.

15. A computer programming method comprising:

describing software-visible physical objects and an instruction set using object-oriented classes; and

identifying when classes are implemented by a computer.

16. A computer programming method comprising:

deriving variable classes from a constant class; and

describing one of the variable classes and the constant class using a single descriptor.

17. A storage medium encoded with machine-readable code, the code including instructions for allowing a computer to implement a computer programming method comprising:  
describing data types as abstract without default or implicit implementations and distinctly from classes or interfaces;

5       describing subtype and supertype relationships between the types; and  
describing representations of values of the types as states of classes of objects with interfaces.

18. The storage medium of claim 17 wherein the method further comprises:  
describing software-visible physical objects and an instruction set using object-oriented

10       classes; and  
identifying when classes are implemented by a computer.

19. The storage medium of claim 17 wherein the method further comprises:  
describing multiple classes of pointer objects, said pointer objects capable of signifying objects.

15       20. The storage medium of claim 17 wherein the method further comprises:  
describing commands for transferring program control in non-sequential ways as routines.

21. The storage medium of claim 17 wherein the method further comprises:  
describing at least one of interfaces, classes, enumerations, subroutines, and other  
20       routines as classes of objects.

22. The storage medium of claim 17 wherein the method further comprises:  
invoking statements in a compilation with results thereof being incorporated into an output of a compiler.

23. The storage medium of claim 17 wherein the method further comprises:  
invoking statements in a compilation to interpret literals in an input of a compiler.

24. The storage medium of claim 17 wherein the method further comprises:

deriving variable classes from a constant class; and

describing one of the variable classes and the constant class using a single descriptor.

25. The storage medium of claim 17 wherein the method further comprises:

deriving variable interfaces from a constant interface; and

describing one of the variable interfaces and the constant interface using a single  
descriptor.

26. The storage medium of claim 17 wherein the method further comprises:

describing formal arguments to routines as objects.

27. The storage medium of claim 17 wherein the method further comprises:

describing formal arguments to routines as a number of arguments including type,

interface, or class of each argument, dataflow attribute of each argument, and preconditions and  
postconditions of routines.

28. The storage medium of claim 17 wherein the method further comprises:

describing subroutines as parameterized classes.

29. A storage medium encoded with machine-readable code for compilation, the code including instructions for causing a computer to implement a method comprising:

- a. generating a description of a computer architecture as a first library, the description including software-visible objects and the instruction set;
- b. implementing a second library of high level objects using the first library; and
- c. binding a source program to implementations in the second library to produce machine instructions dependent on the computer architecture.

30. A storage medium encoded with machine-readable code for re-targeting, the code including instructions for causing a computer to implement a method comprising:

- a. generating a first description of a first computer architecture as a first library, the first description including software-visible objects and instruction set therefor;
- b. generating a second description of a second computer architecture as a second library, the second description including software-visible objects and instruction set therefor;
- c. implementing the first description using the second library to produce a third library; and
- d. binding a source program to implementations in the third library to produce machine instructions dependent on the second computer architecture.

31. A storage medium encoded with machine-readable code, the code including instructions for allowing a computer to implement a computer programming method:  
describing software-visible physical objects and an instruction set using object-oriented classes; and  
5 identifying when classes are implemented by a computer.

32. A storage medium encoded with machine-readable code, the code including instructions for allowing a computer to implement a computer programming method:  
deriving variable classes from a constant class; and  
describing one of the variable classes and the constant class using a single descriptor.

10 33. A signal propagated over a propagation medium, the signal encoded with code including instructions for allowing a computer to implement a computer programming method comprising:

describing data types as abstract without default or implicit implementations and distinctly from classes or interfaces;

15 describing subtype and supertype relationships between the types; and  
describing representations of values of the types as states of classes of objects with interfaces.

34. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

20 describing software-visible physical objects and an instruction set using object-oriented classes; and  
identifying when classes are implemented by a computer.

35. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

describing multiple classes of pointer objects, said pointer objects capable of signifying objects.

5 36. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

describing commands for transferring program control in non-sequential ways as routines.

10 37. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

describing at least one of interfaces, classes, enumerations, subroutines, and other routines as classes of objects.

38. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

15 invoking statements in a compilation with results thereof being incorporated into an output of a compiler.

39. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

invoking statements in a compilation to interpret literals in an input of a compiler.

20 40. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

deriving variable classes from a constant class; and

describing one of the variable classes and the constant class using a single descriptor.



41. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

deriving variable interfaces from a constant interface; and

describing one of the variable interfaces and the constant interface using a single

5 descriptor.

42. The signal propagated over the propagation medium of claim 33 wherein the method further comprises:

describing formal arguments to routines as objects.

43. The signal propagated over the propagation medium of claim 33 wherein the method

10 further comprises:

describing formal arguments to routines as a number of arguments including type, interface, or class of each argument, dataflow attribute of each argument, and preconditions and postconditions of routines.

44. The signal propagated over the propagation medium of claim 33 wherein the method

15 further comprises:

describing subroutines as parameterized classes.

45. A signal propagated over a propagation medium, the signal encode with code for compilation, the code including instructions for causing a computer to implement a method comprising:

- a. generating a description of a computer architecture as a first library, the description including software-visible objects and the instruction set;
- b. implementing a second library of high level objects using the first library; and
- c. binding a source program to implementations in the second library to produce machine instructions dependent on the computer architecture.

46. A signal propagated over a propagation medium, the signal encode with code for re-targeting, the code including instructions for causing a computer to implement a method comprising:

- a. generating a first description of a first computer architecture as a first library, the first description including software-visible objects and instruction set therefor;
- b. generating a second description of a second computer architecture as a second library, the second description including software-visible objects and instruction set therefor;
- c. implementing the first description using the second library to produce a third library; and
- d. binding a source program to implementations in the third library to produce machine instructions dependent on the second computer architecture.

47. A signal propagated over a propagation medium, the signal encoded with code including instructions for allowing a computer to implement a computer programming method comprising:

describing software-visible physical objects and an instruction set using object-oriented

5 classes; and

identifying when classes are implemented by a computer.

48. A signal propagated over a propagation medium, the signal encoded with code including instructions for allowing a computer to implement a computer programming method comprising:

10 deriving variable classes from a constant class; and

describing one of the variable classes and the constant class using a single descriptor.